

# Arnold 5.0+

Below are some commonly asked questions with regards to changes in Arnold 5.

## Where is the Standard shader?

The old *standard* shader has been replaced with the [standard\\_surface](#) shader. The old *standard* shader is still shipped with Arnold as a legacy shader, for compatibility with old scenes. However, it will be removed in a future release.

## Is the Standard Surface shader energy conserving?

If all of the individual weights and colors are less than or equal to 1.0, then the [standard\\_surface](#) shader is energy conserving. Unlike the old Standard shader, you don't need to worry about the sum of weights being less than 1.0 or manually enable Fresnel, which is always enabled.

## Where is 'Refraction Roughness' in the Standard Surface shader?

The [specular\\_roughness](#) affects both specular reflection and refraction. There is also an [extra\\_roughness](#) parameter in [transmission](#) to add some additional roughness for refraction. You can, however, use [coat](#) to create a rough reflection layer over a sharp refraction.

## Where is the specular GGX microfacet distribution?

GGX is now the default model for the [standard\\_surface](#) shader.

## How can I control the index of refraction for transparent materials?

You can use the [Specular IOR](#) attribute in the same way to handle transparent materials such as glass, diamond, etc.

## How can I control Backlighting/Translucency in the Standard Surface shader?

Enabling [thin\\_walled](#) and setting [subsurface\\_weight](#) to, for example, 0.5 to have half the light reflected and half transmitted.

## IOR and Metalness appear to do the same thing. Why use one over the other?

Using very high IOR values can look quite similar to metalness. It looks the same if you set base color to the specular color and the specular color to black. The difference is that you get an extra reflection at the edges, with the specular color controlling the edge tint. The metal Fresnel works the same as in the [Complex IOR](#) shader, with the artistic parameters.

## Where is the Skin shader?

You should use the [subsurface](#) parameter of the [Standard Surface](#) shader. It uses the high-quality empirical SSS profile, which preserves sharp geometric and bump detail without the need for three layers of varying depth, as the old skin shader did.

## Where is the Hair shader?

Use the new [standard\\_hair](#) shader instead. The old *hair* shader is shipped with Arnold as a legacy shader, for compatibility with old scenes. However, it will be removed in a future release.

## Where have the UV parameters gone?

The [standard\\_hair](#) shader does not have a "uvset" parameter at all. Rather all image and procedural texture nodes have a "uvset" parameter instead so that they can pick up the appropriate texture coordinates there.

## Where is the Volume Collector shader?

Use the new [standard\\_volume](#) shader instead. The old *Volume Collector* shader is shipped with Arnold as a legacy shader, for compatibility with old scenes. However, it will be removed in a future release.

## How can I use OSL shaders?

Information about OSL shaders can be found [here](#).

## Where is the Shadow Catcher shader?

The [shadow\\_catcher](#) shader is now called the [shadow\\_matte](#) shader.



### ASS Files

Although the syntax of the file format hasn't changed at all, .ass files generated with Arnold 4 may not render correctly with Arnold 5 and so should be avoided. Likewise, .ass files generated with Arnold 5 may not render correctly with Arnold 4.

### Sky Shader

The sky shader is deprecated and will be removed from a future release. You should instead use the visibility settings in the [Skydome Light](#).

### Skydome Light

The [camera](#) visibility in the skydome light replaces the need for creating a sky shader and carefully setting its visibility attributes to avoid double counting. The Skydome light shows up in the indirect AOVs, not the direct. It can be changed to the direct light AOV by setting the light's AOV to "default" to place it in the default light group, or another name for some other light group.